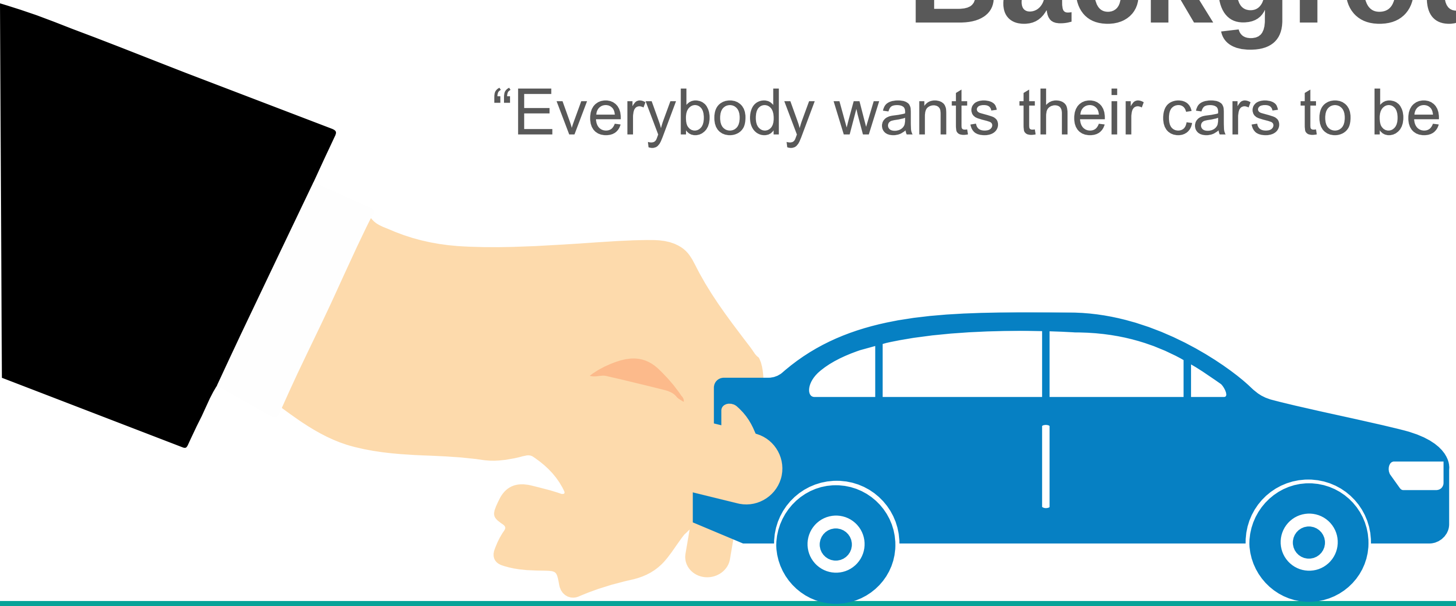
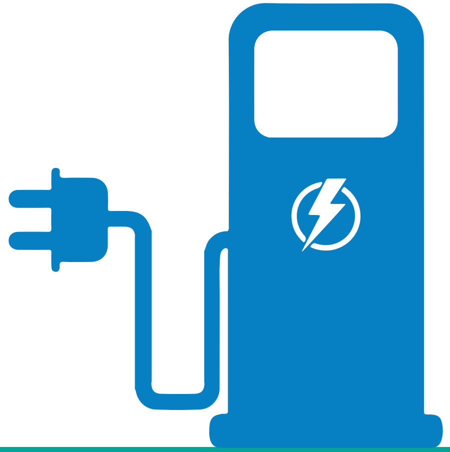
Used Car Pricing Prediction

Group 22: Jiachen Liu and Thinh Nguyen

Background

"Everybody wants their cars to be sold at an ideal price"





Pricing Model has been widely used in many fields.

We will implement car value regression model to predict the ideal price of a used car

Fun Facts:

$36.7 mil
Market Size
2020

1.6 mil
Vehicles

30%»
Demand
Spring 2021

Primary Steps:

- Exploratory Data Analysis
- Feature Engineering (PCA)
- Classical Models (L1, L2, etc.)
- Advanced Models (XGBoost)
- Model Evaluation

Context

1 Data Source

Kaggle

10781 Sample Size

BMW Dataset
Csv file

6 Techniques

Regression Model

1 Geography

United Kingdom

3 Categorical

Model, Transmission,
Fuel Type

6 Numerical

Year, Price, Mileage, Tax,
Mpg, Engine Size

Exploring Metadata

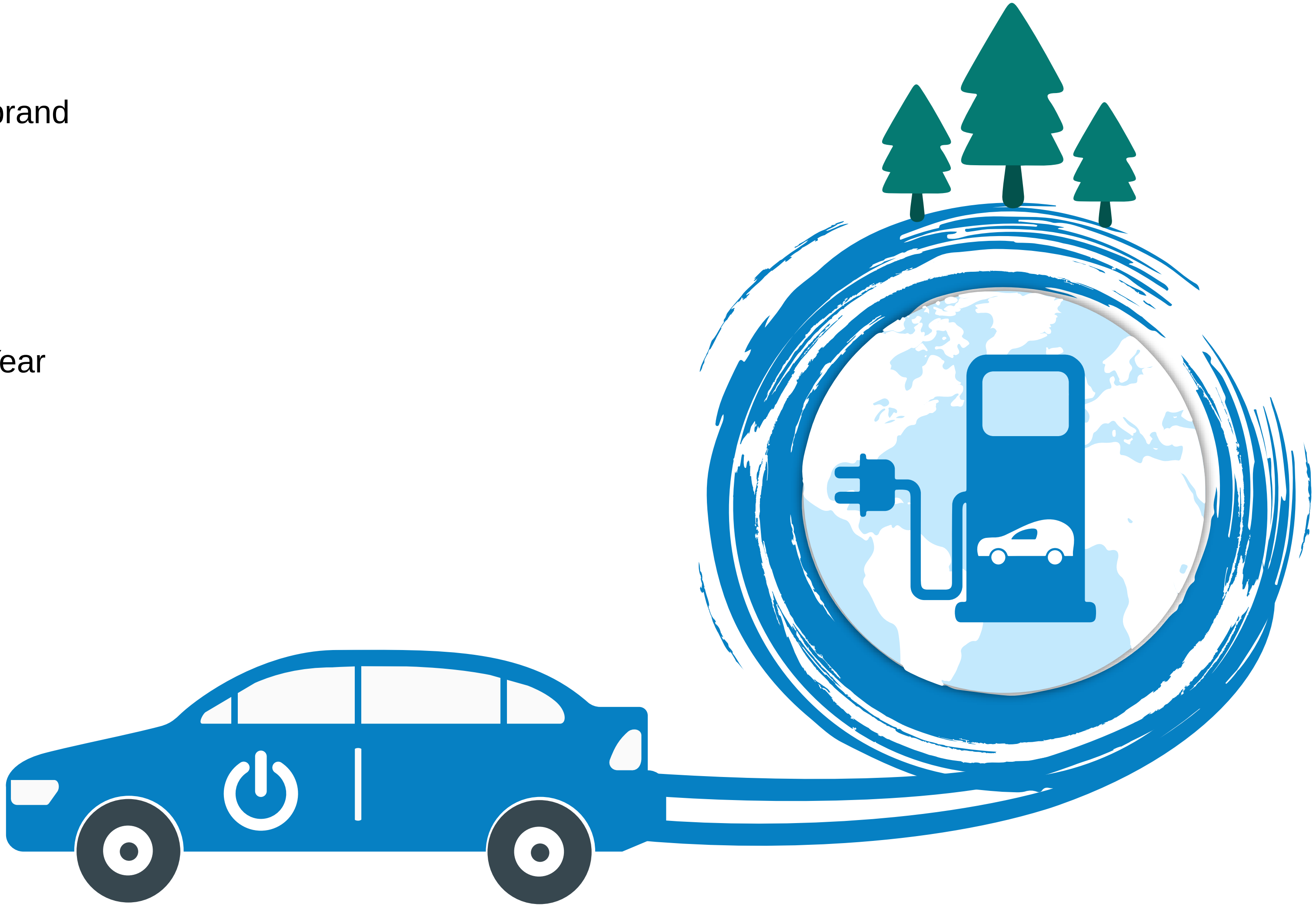
3 Categorical Features

Model	Car Model of the brand
Transmission	Type of gearbox
fuelType	Engine Fuel

6 Numerical Features

Year	Car Registration Year
Price	Car Price (in £)
Mileage	Distance Used
Tax	Road Tax (in £)
Mpg	Miles per gallon
engineSize	Size in liters

Used Car
Pricing Prediction

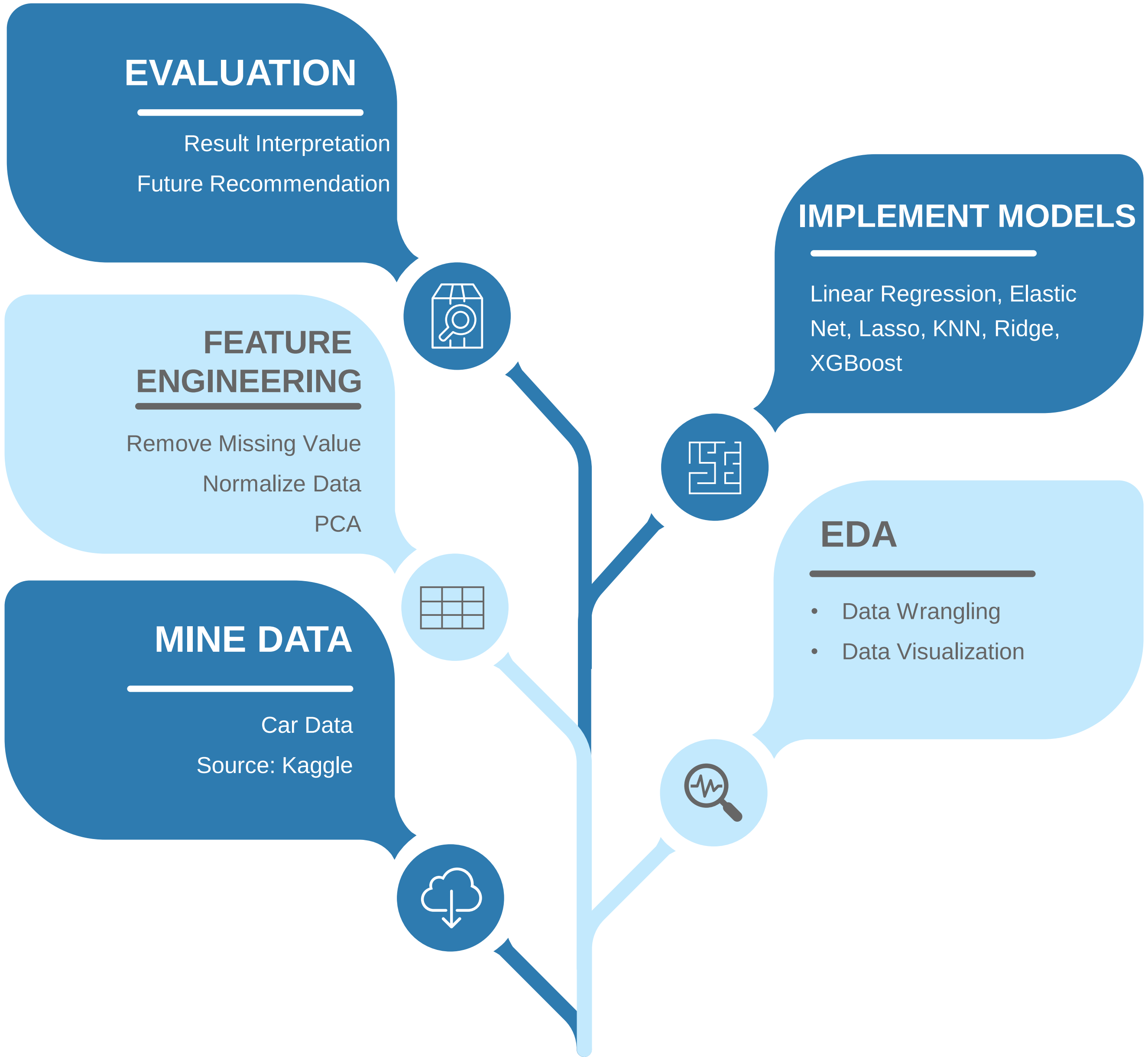


Exploring Data

Sample

Model	Year	Price (in £)	Transmission	Mileage	fuelType	Tax (in £)	Mpg	engineSize (in liters)
5 Series	2014	11,200	Automatic	67,068	Diesel	125	57.6	2
2 Series	2018	16,250	Manual	10,401	Petrol	145	52.3	2
1 Series	2015	15,499	Semi-Auto	20,000	Diesel	125	60.1	2
I3	2015	17,400	Automatic	29,465	Electric	0	470.8	1
X5	2016	34,498	Automatic	17303	Hybrid	140	113	1.5
3 Series	2017	14,250	Automatic	55594	Other	135	148.7	2
M4	2020	50,000	Semi-Auto	700	Petrol	145	34	3

Flow Chart



EVALUATION
Result Interpretation
Future Recommendation
FEATURE ENGINEERING
Remove Missing Value
Normalize Data
PCA
MINE DATA
Car Data
Source: Kaggle
IMPLEMENT MODELS
Linear Regression, Elastic Net, Lasso, KNN, Ridge, XGBoost
EDA
• Data Wrangling
• Data Visualization

Exploratory Data Analysis

Remaining Input Variables:

Model

Year

Transmission

Mileage

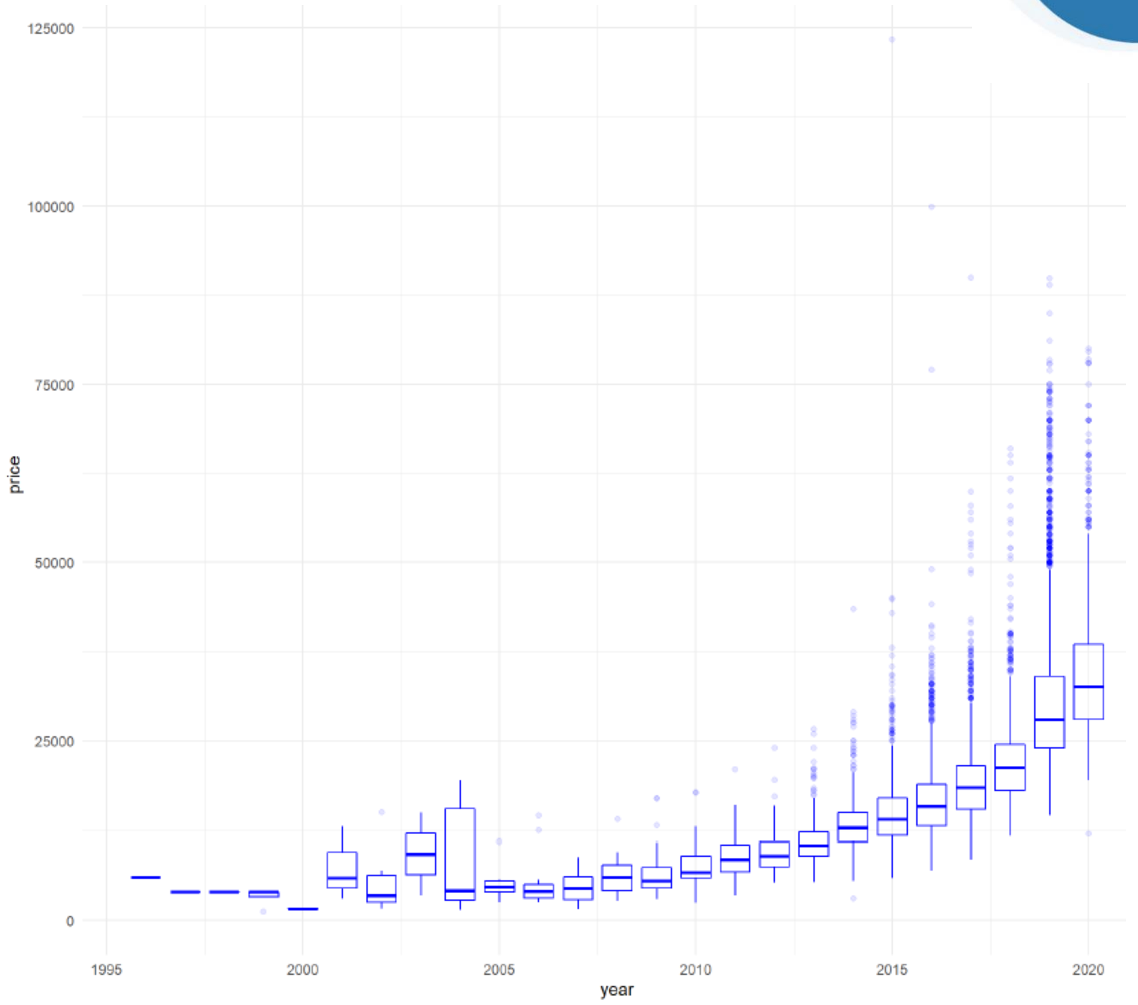
fuelType

Tax

Mpg

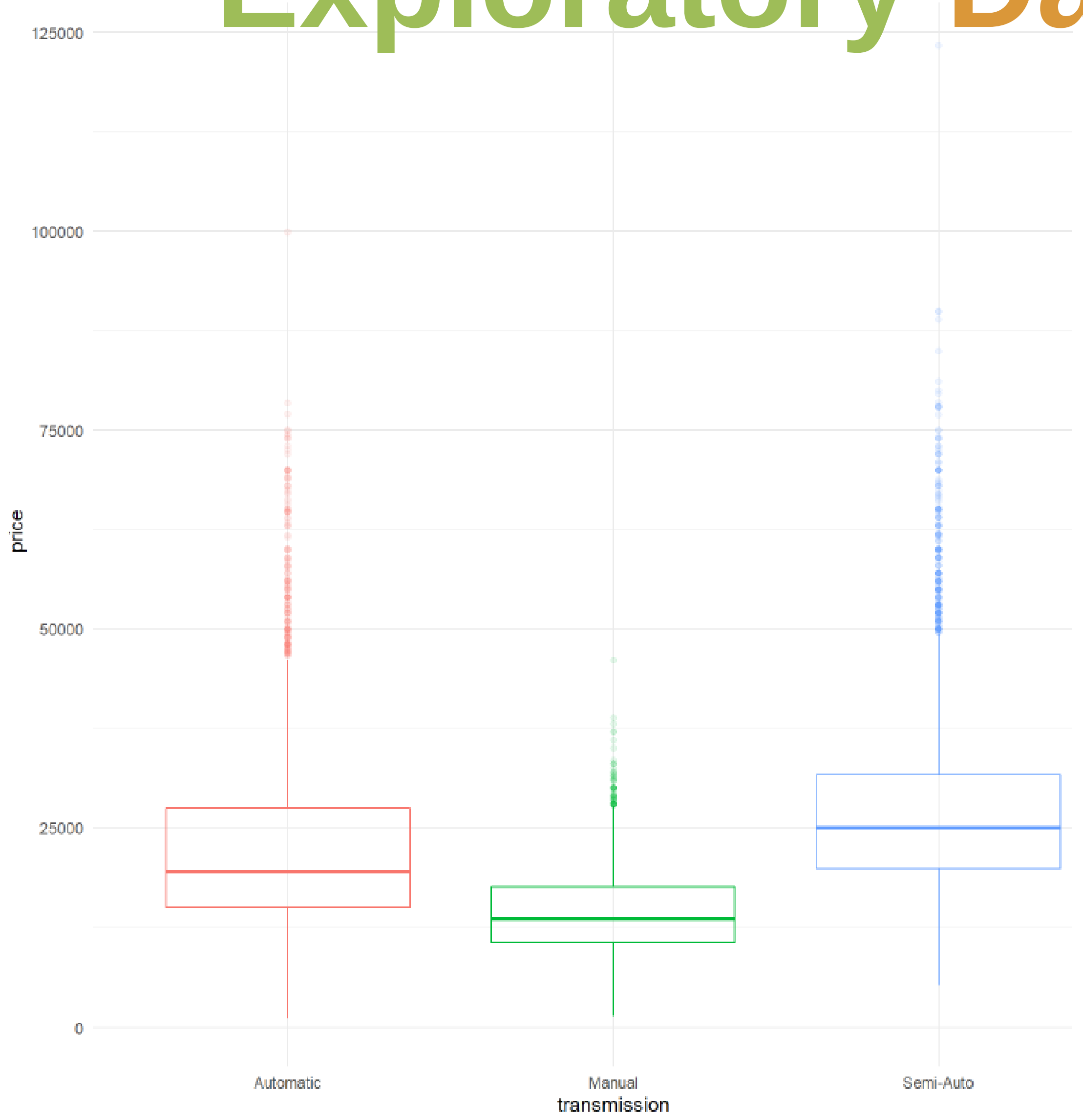
engineSize

Target Variable: **Price**

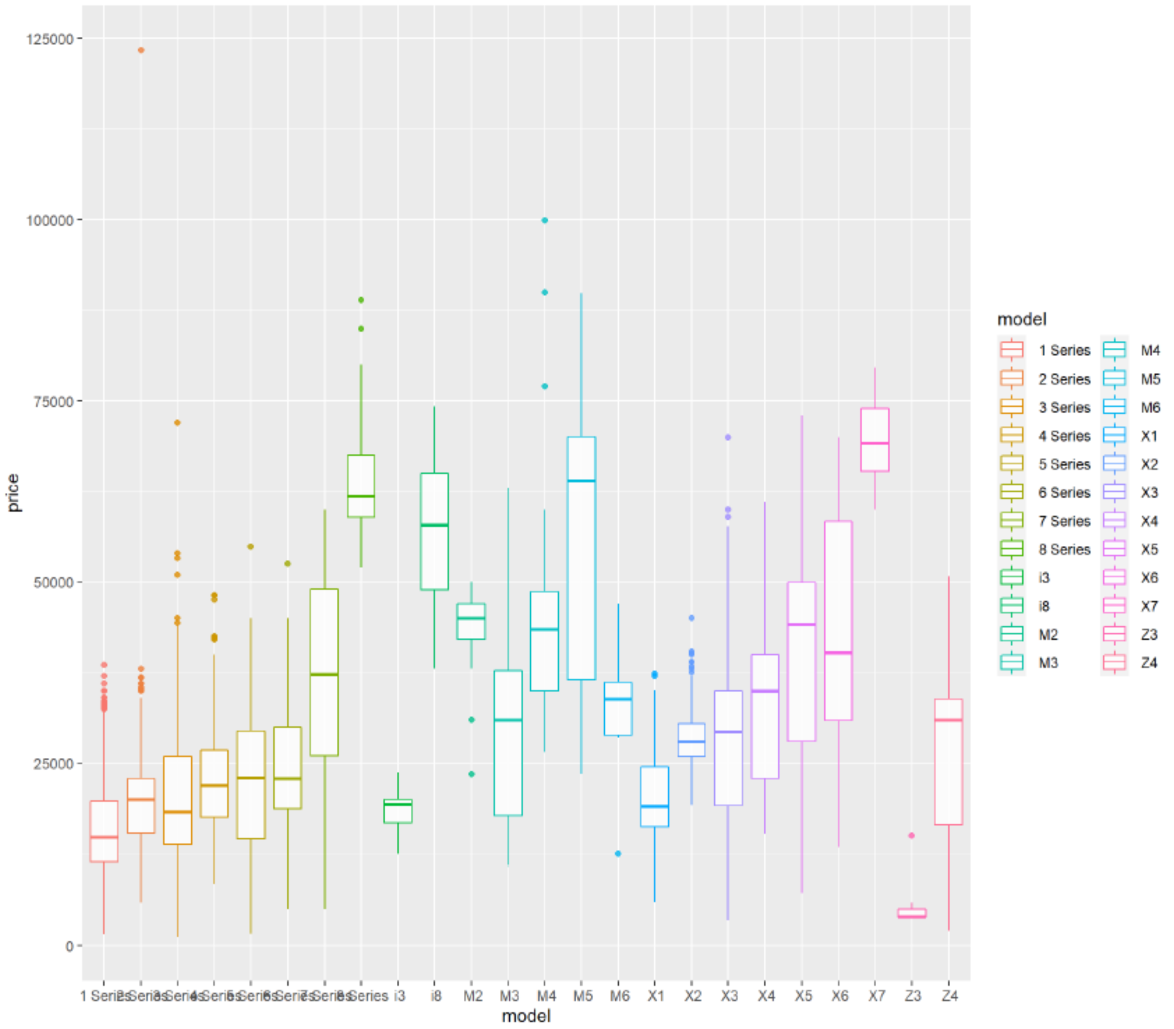


Relationship between **Year** and **Price**

Exploratory Data Analysis



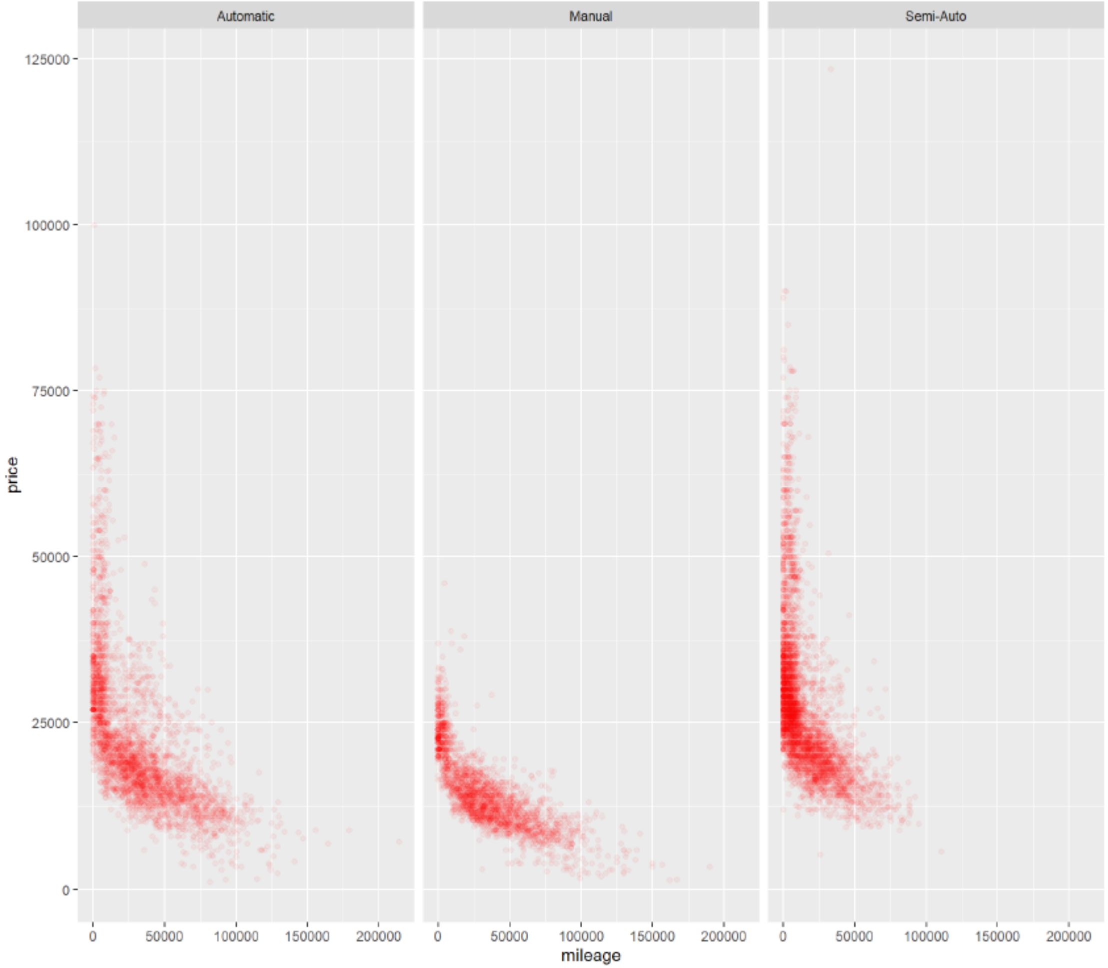
Boxplot by Transmission



Boxplot by Model

Exploratory Data Analysis

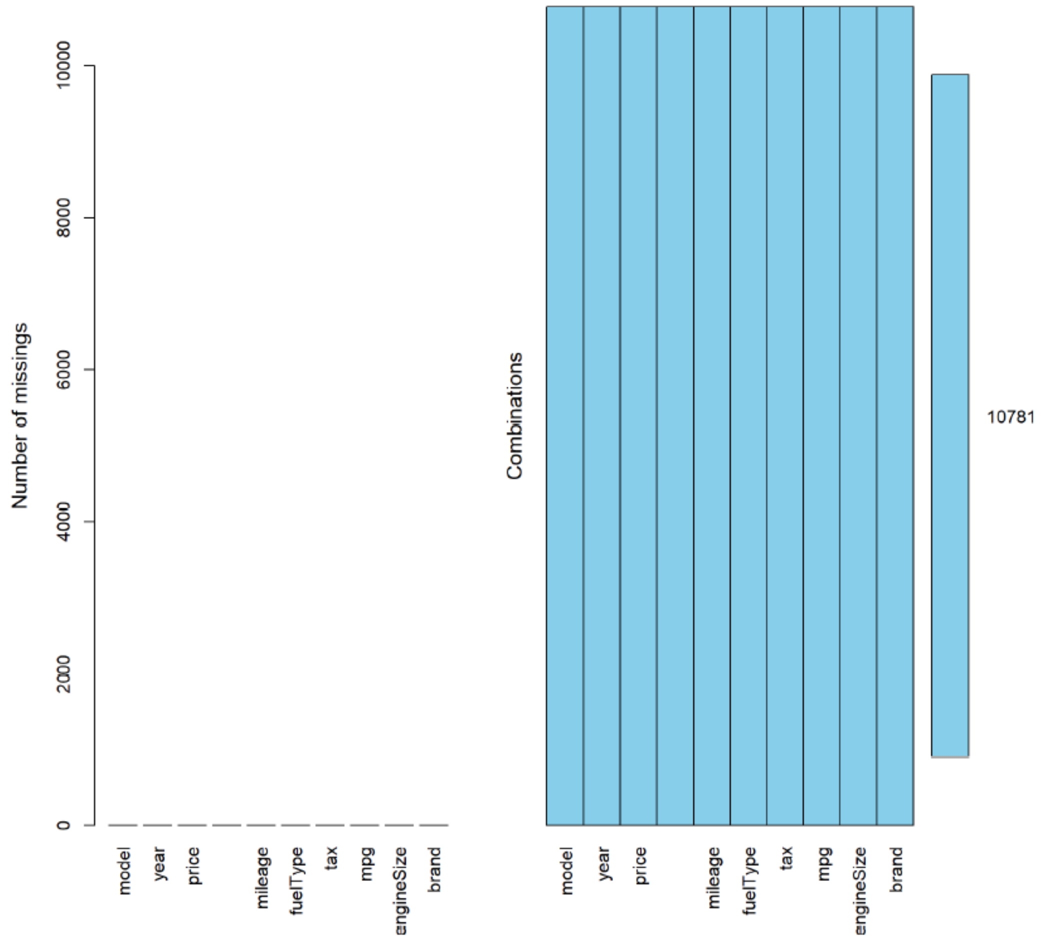
Relationship
between
mileage
and price
(Transmission)



Relationship between mileage and price
(2015-2020)

Feature Engineering

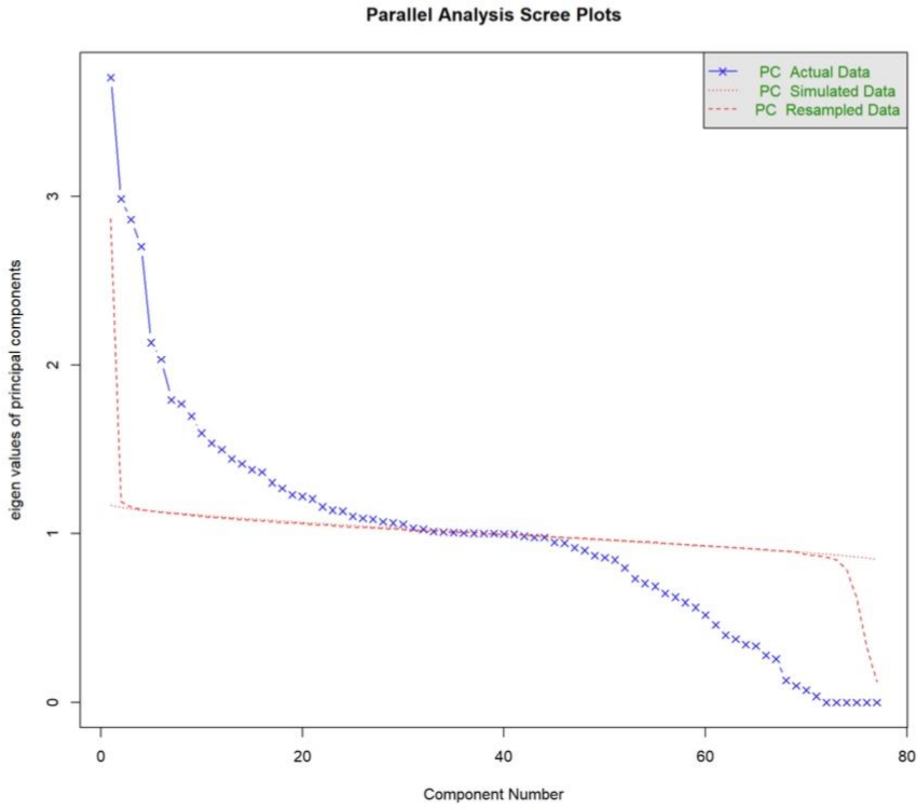
Check **Missing** value



Encode Categorical Data (One-Hot Encoding)

Normalize Data

Perform **PCA** and choose appropriate **Number of Components**



From these plots , we can see there is no missing value, and the number of component is 31.

Model Implementation

Implement Different Regression Models

Linear Regression

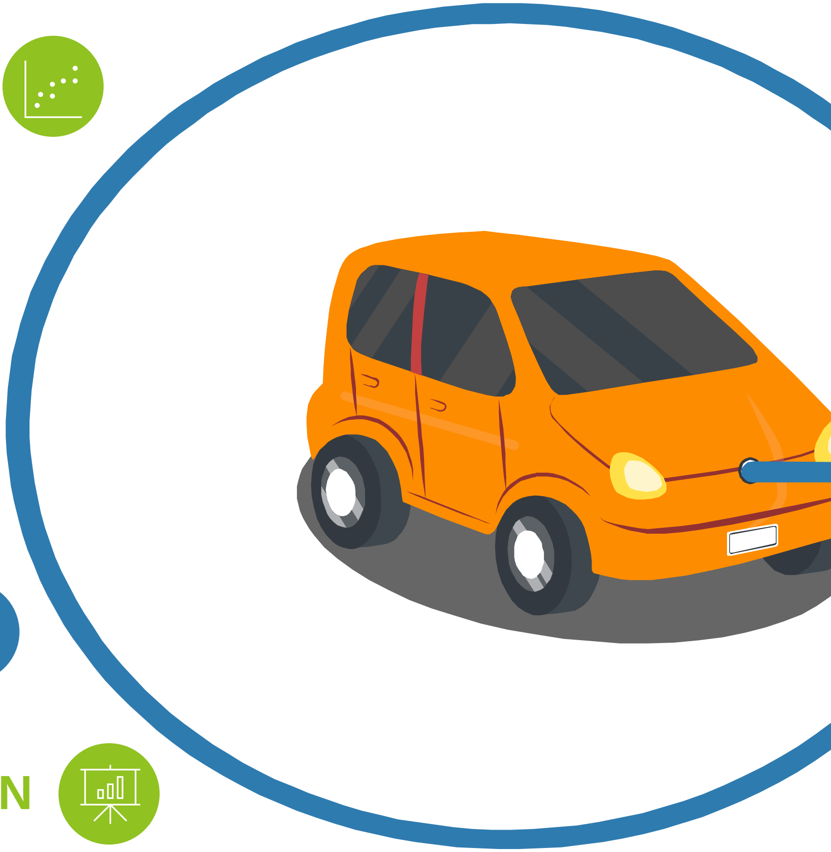
Lasso Regression

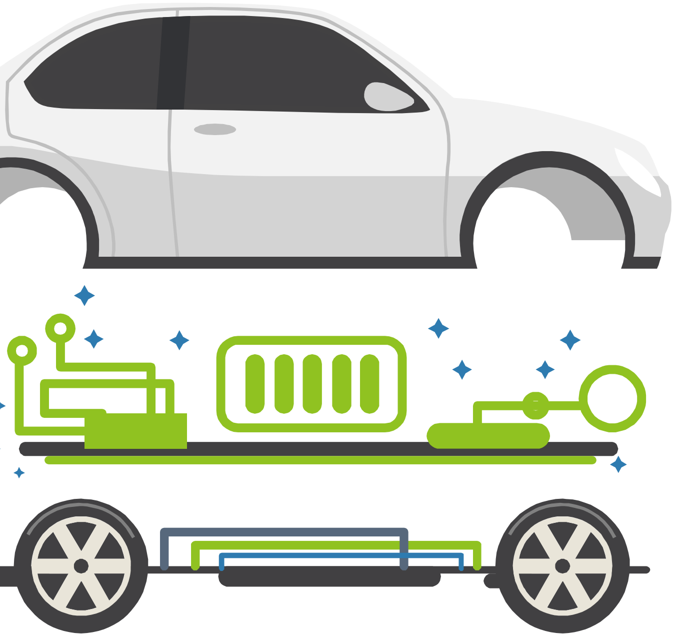
Ridge Regression

Elastic Net

KNN

XGBoost





Splitting Dataset

After PCA

Train Dataset

80%

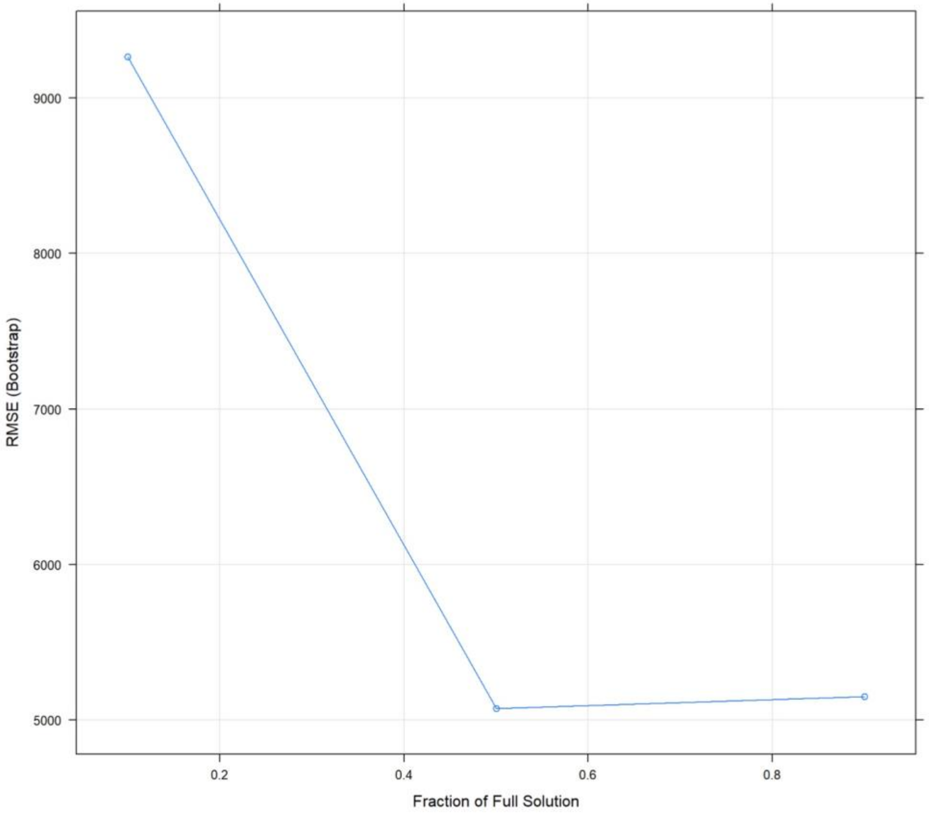
Test Dataset

20%

Linear Regression

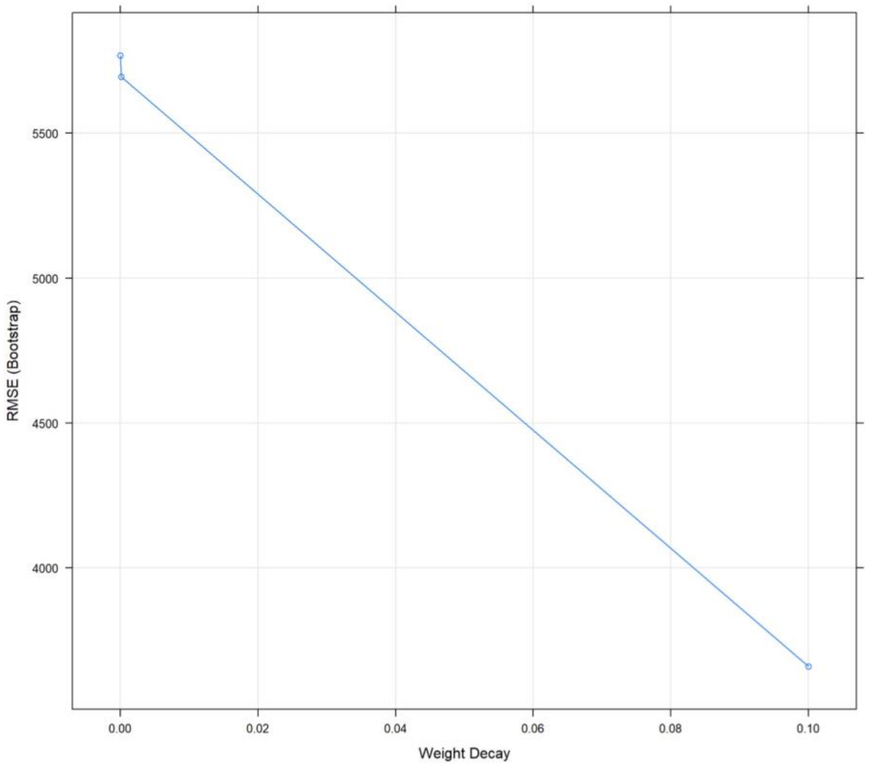
```
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 22737.80      39.77 571.682  < 2e-16 ***
## PC1          2124.82      10.83 196.247  < 2e-16 ***
## PC2          1766.52      13.43 131.515  < 2e-16 ***
## PC3           868.48      14.30  60.747  < 2e-16 ***
## PC4            55.16      14.62   3.772 0.000163 ***
## PC5          -676.06      18.69 -36.164  < 2e-16 ***
## PC6          -953.21      19.79 -48.164  < 2e-16 ***
## PC7          -648.42      21.97 -29.510  < 2e-16 ***
## PC8           -34.48      20.54  -1.678 0.093343 .
## PC9          -189.38      23.40  -8.093 6.62e-16 ***
## PC10         -489.52      40.16 -12.189  < 2e-16 ***
## PC11         1382.27      27.53  50.219  < 2e-16 ***
## PC12          319.12      30.81  10.359  < 2e-16 ***
## PC13          -50.47      26.20  -1.926 0.054140 .
## PC14          103.12      29.97   3.440 0.000584 ***
## PC15          900.41      29.68  30.341  < 2e-16 ***
## PC16          600.78      29.73  20.206  < 2e-16 ***
## PC17          226.99      30.58   7.423 1.26e-13 ***
## PC18          619.40      32.29  19.185  < 2e-16 ***
## PC19          544.67      32.32  16.850  < 2e-16 ***
## PC20          365.94      34.76  10.529  < 2e-16 ***
## PC21         1293.34      33.67  38.409  < 2e-16 ***
## PC22         -145.73      33.92  -4.297 1.75e-05 ***
## PC23           42.33      37.66   1.124 0.261097
## PC24          143.17      35.54   4.028 5.66e-05 ***
## PC25        -1039.04      36.20 -28.704  < 2e-16 ***
## PC26          686.64      36.51  18.805  < 2e-16 ***
## PC27          -11.53      36.66  -0.315 0.753064
## PC28          180.31      36.96   4.879 1.09e-06 ***
## PC29          -95.96      37.13  -2.584 0.009773 **
## PC30          569.36      37.91  15.019  < 2e-16 ***
## PC31         -330.95      37.44  -8.839  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 3649 on 8390 degrees of freedom
## Multiple R-squared:  0.897,  Adjusted R-squared:  0.8966
## F-statistic:  2356 on 31 and 8390 DF,  p-value: < 2.2e-16
```

Lasso Regression



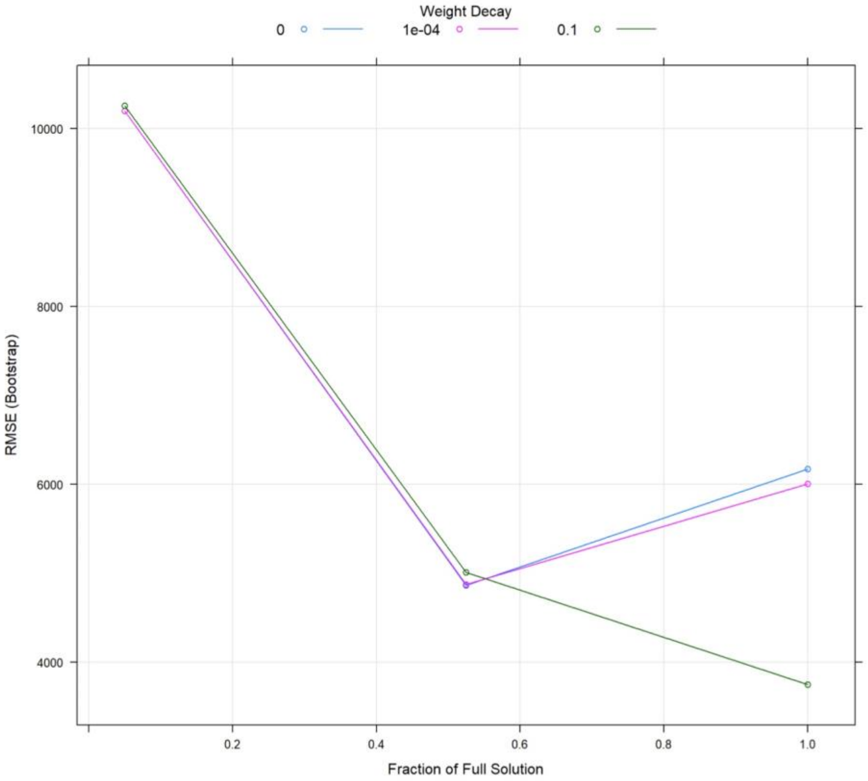

```
## The lasso
##
## 8422 samples
##   31 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 8422, 8422, 8422, 8422, 8422, 8422, ...
## Resampling results across tuning parameters:
##
##   fraction  RMSE      Rsquared   MAE
##   0.1       9266.520  0.5580142  6736.382
##   0.5       5073.275  0.8236085  3533.407
##   0.9       5151.493  0.8585858  2688.957
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was fraction = 0.5.
```

Ridge Regression



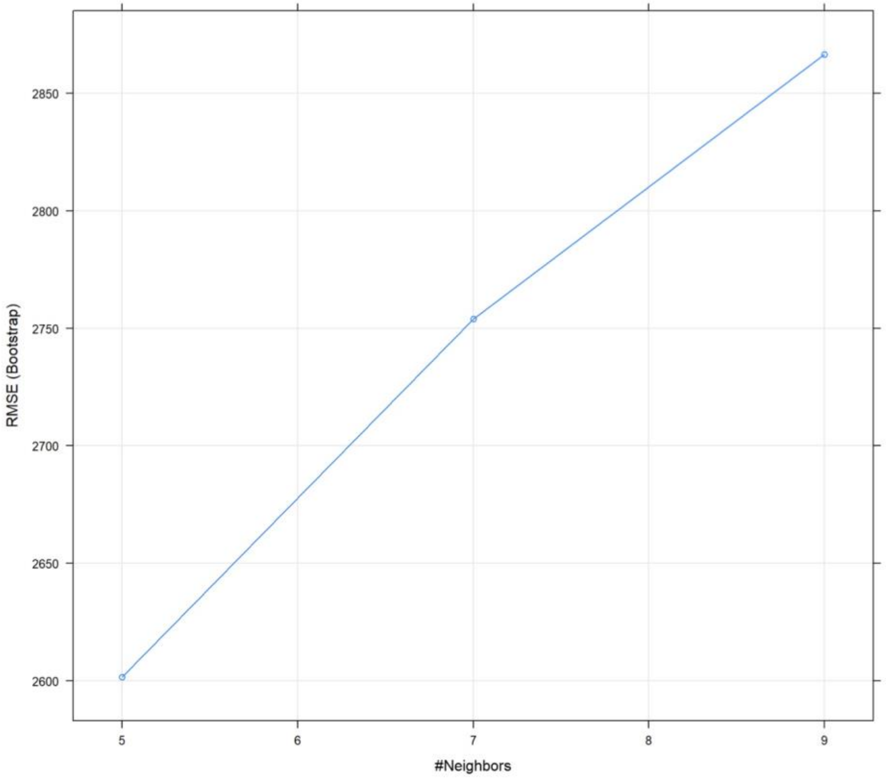

```
## Ridge Regression
##
## 8422 samples
##   31 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 8422, 8422, 8422, 8422, 8422, 8422, ...
## Resampling results across tuning parameters:
##
##   lambda  RMSE      Rsquared   MAE
##   0e+00   5769.355  0.8297025  2678.747
##   1e-04   5695.735  0.8303368  2676.700
##   1e-01   3659.588  0.8951695  2613.910
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was lambda = 0.1.
```

Elastic Net



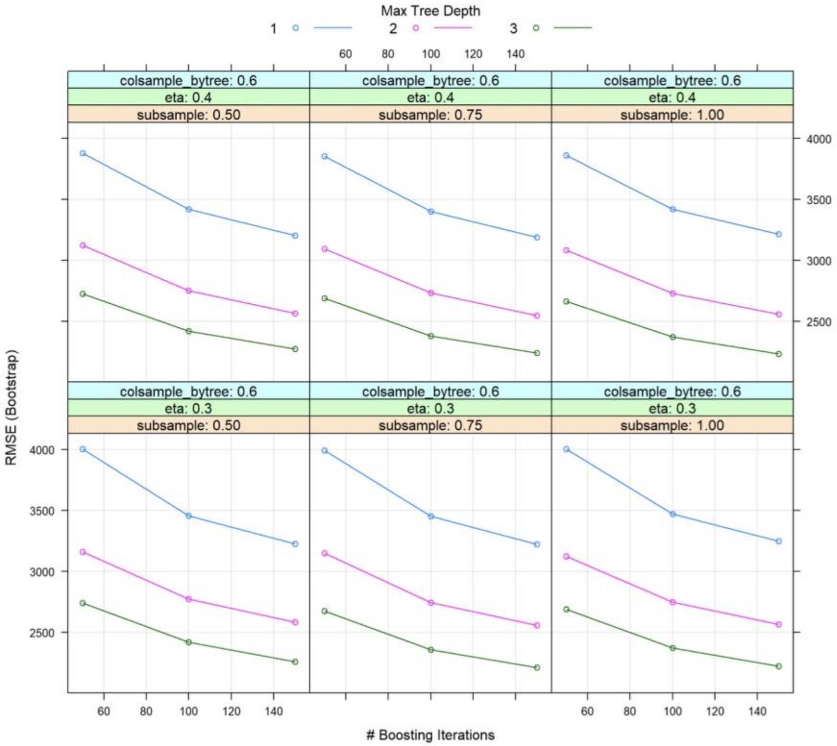

```
## Elasticnet
##
## 8422 samples
##   31 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 8422, 8422, 8422, 8422, 8422, 8422, ...
## Resampling results across tuning parameters:
##
##   lambda  fraction  RMSE       Rsquared   MAE
##   0e+00   0.050     10198.447  0.4622664  7482.738
##   0e+00   0.525      4867.566  0.8348269  3395.957
##   0e+00   1.000      6172.367  0.8051181  2732.327
##   1e-04   0.050     10202.661  0.4613966  7486.154
##   1e-04   0.525      4874.250  0.8345093  3400.482
##   1e-04   1.000      6009.335  0.8067995  2725.737
##   1e-01   0.050     10260.675  0.4600051  7532.487
##   1e-01   0.525      5011.316  0.8272891  3490.770
##   1e-01   1.000      3749.853  0.8907413  2629.293
##
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were fraction = 1 and lambda = 0.1.
```

KNN Regression




```
## k-Nearest Neighbors
##
## 8422 samples
##   31 predictor
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 8422, 8422, 8422, 8422, 8422, 8422, ...
## Resampling results across tuning parameters:
##
##   k  RMSE      Rsquared   MAE
##   5  2601.653  0.9470779  1231.675
##   7  2753.984  0.9408119  1348.401
##   9  2866.505  0.9359094  1446.883
##
## RMSE was used to select the optimal model using the smallest value.
## The final value used for the model was k = 5.
```

XGBoost



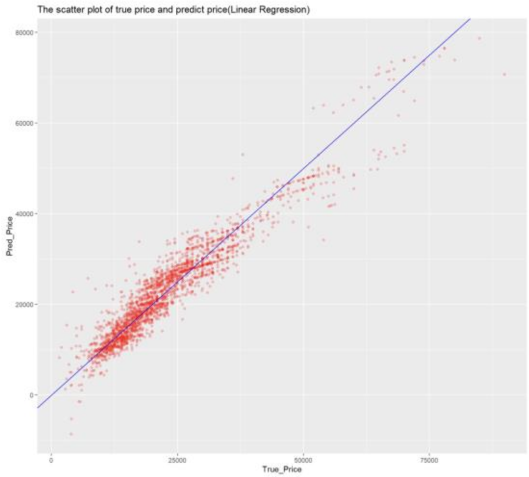

```
##  parameter 'min_child_weight' was held constant at a value of 1
## RMSE was used to select the optimal model using the smallest value.
## The final values used for the model were nrounds = 150, max_depth = 3, eta
##  = 0.4, gamma = 0, colsample_bytree = 0.8, min_child_weight = 1 and subsample
##  = 1.
```

RMSE	R^2	MAE
2133.722	0.9649395	1254.251

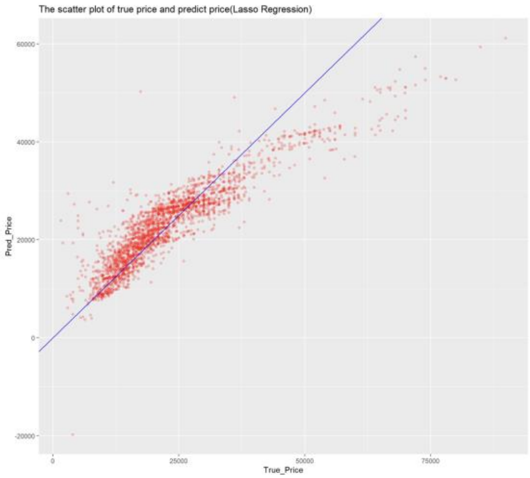
Model Evaluation

- **Implement** each model on test set
- **Plot** the prediction price and true price in test set
- **Evaluate** the RMSE , R^2 and MAE of each model

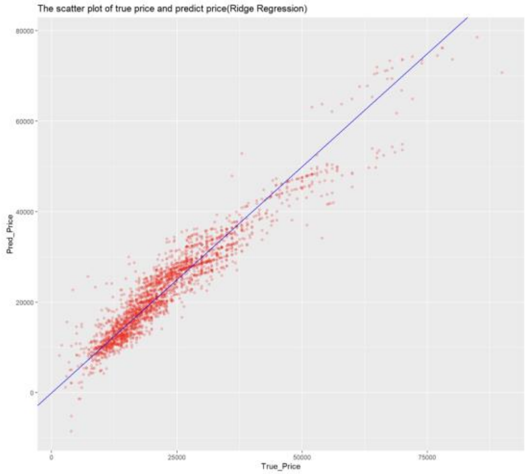
Prediction Price and True Price



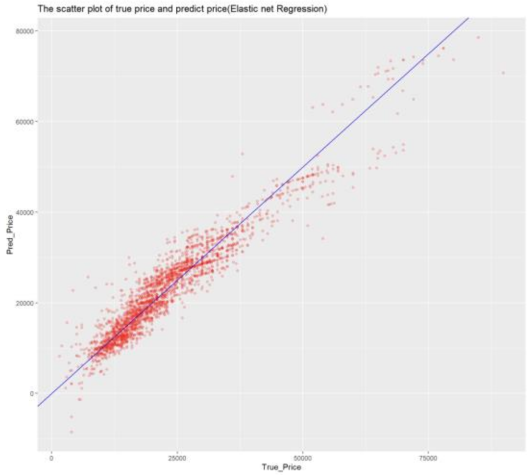
Linear Regression



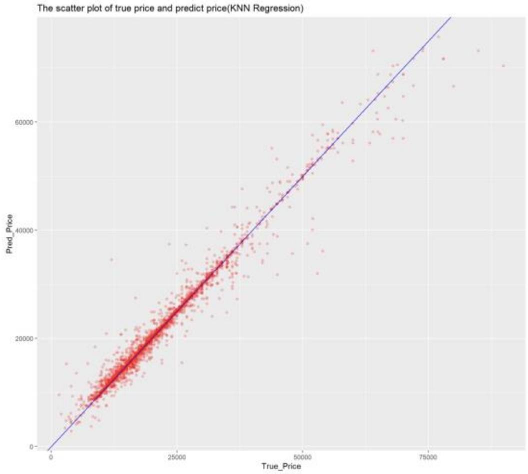
Lasso Regression



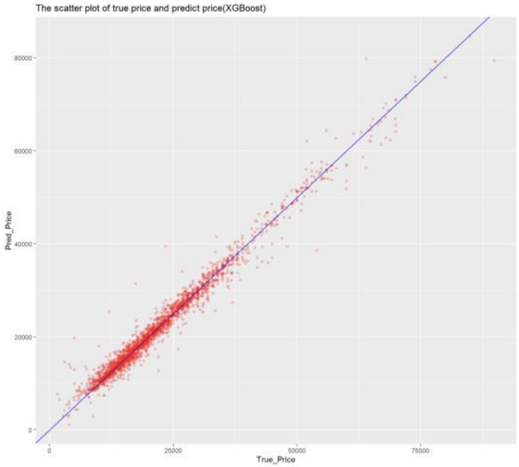
Ridge Regression



Elastic Net



KNN



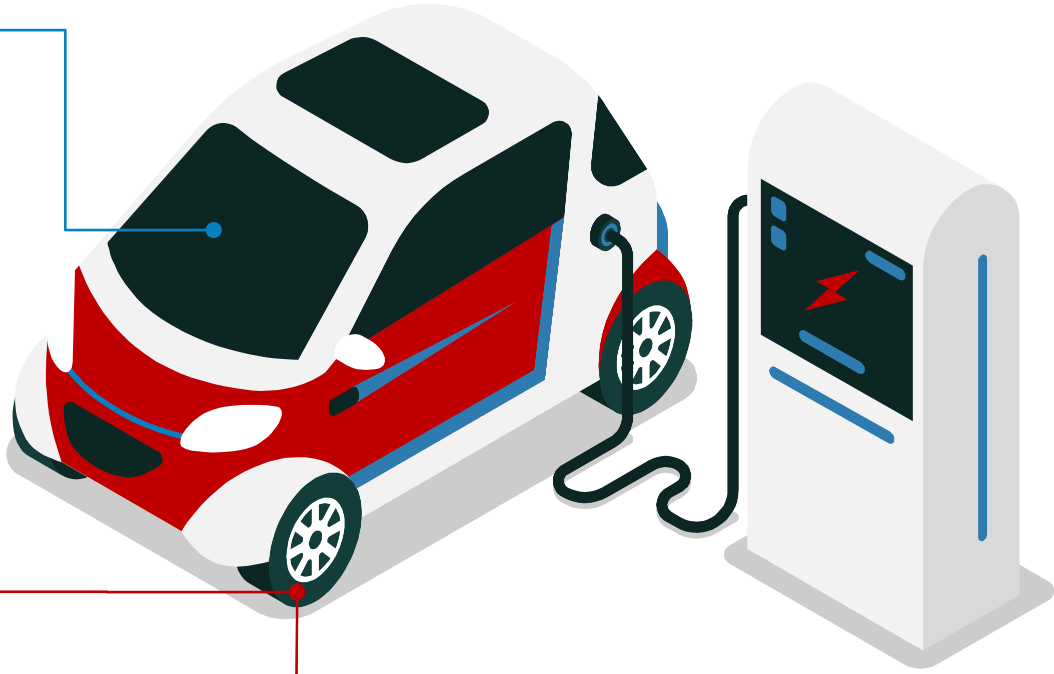
XGBoost

	RMSE	R^2	MAE
Linear Regression	3535.540	0.9081	2583.909
Lasso Regression	5256.004	0.82924	3593.149
Ridge Regression	3521.634	0.90883	2578.9292
Elastic Net	3521.6345	0.90883	2578.9292
KNN	2107.9561	0.96754	1056.1253
XGBoost	1848.3558	0.97491	1179.9157

Discussion

Best Model

XGBoost
R^2: 97.49%



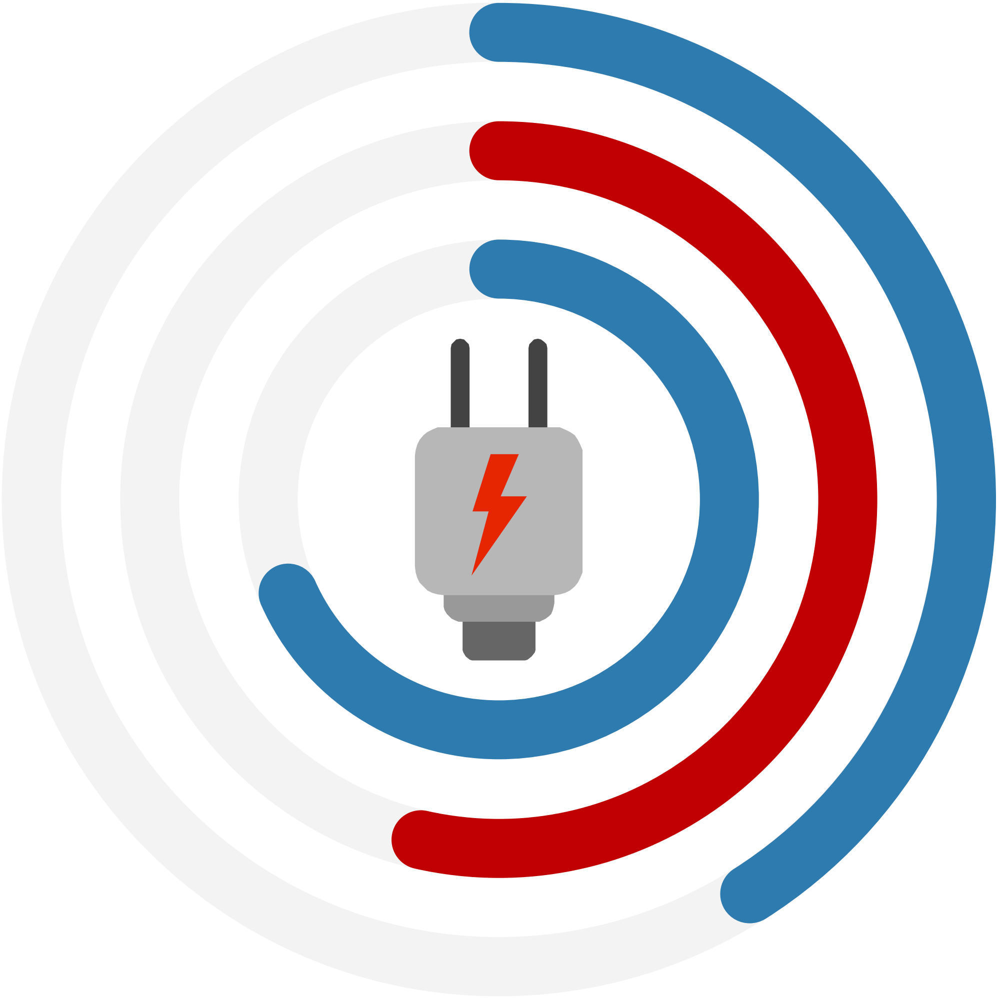
Worst Model

Lasso
R^2: 82.92%

Lasso's Underpderformance

- Use **L1 norm** as penalty
- L1 norm cause some feature weight to **0**
- Cause **Overfitting**

Future Work



Try

More

Model such as Bayesian Ridge Regression

Try

Other

Dimension Reduction Method such as LDA

Compare

Result

Between Dataset with PCA and Dataset without PCA

THANK YOU